Személynév-egyértelműsítés a magyar weben

Nagy T. István[1], Farkas Richárd[2]

[1] Szegedi Tudományegyetem, Informatikai Tanszékcsoport
6720, Szeged, Árpád tér 2.
nistvan@inf.u-szeged.hu
[2] MTA-SZTE Mesterséges Intelligencia Kutatócsoport
Szeged, Tisza Lajos krt. 103. III. lépcsőház
rfarkas@inf.u-szeged.hu

Kivonat: Ebben a cikkben bemutatjuk saját személynév-egyértelműsítő rendszerünket, amely képes egy adott névhez mint keresőkifejezéshez tartozó weboldalakból a különböző személyek és a hozzájuk tartozó honlapok azonosítására. Ezen megközelítés alapvetően az egyes személyekhez automatikusan felismert bibliográfiai jellemzők segítségével rendeli a különböző emberekhez az egy névhez tartozó honlapokat. Tehát a klaszterezés során nem használtuk fel az egyes weboldalak teljes tartalmát. Továbbá reprezentáljuk a magyar személynév-egyértelműsítő korpuszunkat is, melyen kiértékeltük rendszerünket. A kiértékelésre a BCubed metrikákat alkalmaztuk.

1 Bevezetés

Az internetfelhasználók egyik leggyakoribb tevékenysége személyek vagy hozzájuk kapcsolódó információk keresése az interneten. A keresőkben használt keresőkifejezések csaknem 30%-a tartalmaz valamilyen személynevet [1]. Viszont a nevek nagymértékben többértelműek: az amerikai népszámlálási hivatal szerint csaknem 100 millió emberhez alig 90.000 különböző név tartozik [2]. Ugyanez igaz hazánkban is, hiszen a 9 leggyakoribb családnév több mint négymillió ember családneve [3]. Ebből kifolyólag ezen keresőkifejezések eredményei az azonos nevű, de különböző személyekhez tartozó honlapokat tartalmaznak.

A személynevek egyértelműsítése több szempontból is kihívásokkal teli (speciális jelentés-egyértelműsítési) feladat. Egyrészt előfordulhat, hogy az egyes nevek többértelműek, több ezer embernek lehet azonos utó- és/vagy vezetékneve Másrészt bizonyos nevek rendkívül változékonyak, így előfordulhat, hogy egy személyhez tartozó nevet többféleképpen is leírhatunk.

Az egy adott névhez tartozó honlapok különböző személyek szerinti klaszterezésének feladatát a 2007-ben először megrendezésre kerülő Web People Search nyílt nemzetközi verseny tűzte ki céljául [4]. A rendszerek kiértékelése során a szervezők arra a következtetésre jutottak, hogy az egyes honlapok személyekhez való rendelése során igen hasznos jellemzőknek bizonyultak a személyekhez tartozó különböző bibliográfiai attribútumok [5]. Ebben a cikkben bemutatjuk a magyar személynév-

egyértelműsítő korpuszunkat[1] és egy olyan rendszert, amely alapvetően az egyes személyekhez automatikusan felismert bibliográfiai jellemzők segítségével rendeli a különböző emberekhez az egy személyhez tartozó honlapokat. Tehát a klaszterezés során nem használtuk fel az egyes weboldalak teljes tartalmát.

A feladat megoldása során 16 különböző jellemzőt azonosítottunk automatikusan úgymint: *családtag*, *mentor*, *egyéb név*, *iskola*, *díj*, *affiliáció*, *e-mail*, *telefonszám*, *fax*, *weboldal*, *születési dátum* és *hely*, *foglalkozás*, *diplomafokozat, nemzetiség.* Ekkor egy adott oldalt a kinyert jellemzők által leírt vektor reprezentált, melyben az egyes jellemzőket fontosságuk szerint súlyoztuk. Ezután definiáltunk egy hasonlósági mértéket, majd egy csoportba rendeltük a hasonló dokumentumokat.

Az angol és magyar nyelvű személynév-egyértelműsítő rendszerünk kiértékelése azt mutatja, hogy megközelítésünk eredményei szignifikánsan jobbak, mint a klasszikus dokumentumklasszifikációs megközelítéseké.

2 Kapcsolódó munkák

A webtartalom-bányászat célja az interneten elérhető szöveges dokumentumokból valamilyen szempont szerint hasznosnak vélt információk kinyerése. A fejlődés motorja a pénzügyi haszon, hiszen a kibányászhatatlannak vélt, vagy csak nagyon erőforrás-igényesen elérhető információk, összefüggések nagyon sokat érhetnek.

A kezdeti klasszikus webtartalom-bányászati próbálkozások 1998-'99 környékén jelentek meg [4, 5]. Ezek az alapvetően szabályalapú rendszerek vagy kézzel előállított szabályokon, vagy egy manuálisan annotált korpusz felügyelt tanulása során előálló szabályokon alapultak. A következő generációs megközelítések alapvetően gyengén felügyelt tanulási módszerek voltak. Ekkor a különböző rendszerek inputja egy lista volt célinformáció-párokkal. Ezen rendszerek célkitűzése, hogy összegyűjtsék azokat a párokat, amelyek kapcsolódnak egymáshoz. Ilyen párok lehettek például összefüggő entitások, mint ország – főváros [6], híres emberek és kapcsolataik [7], vagy entitás – attribútum párok, mint Nobel díjazottak – év [8]. Ezen rendszerek általában letöltötték azokat a honlapokat, amelyek tartalmazták az aktuális párokat, majd szintaktikai/szemantikai szabályokat tanultak azok mondataiból. Végül egy új weboldalkorpuszon alkalmazták az előzetesen megtanult mintákat, hogy új párokat nyerjenek ki. Ezen megközelítések alapvetően az internet redundanciáját használják ki. Azon a hipotézisen alapulnak, mely szerint az interneten a hasznos információk több helyen is elérhetőek, ezért néhány nagyon pontos szabály segítségével a szükséges információk kinyerhetővé válnak.

A második WePS kampány személynév-egyértelműsítési részfeladatán a beküldött rendszerek [5] többsége használt valamilyen előfeldolgozó lépést, mielőtt az egyes dokumentumokat reprezentálták volna. Majd valamilyen általános klaszterező algoritmust alkalmaztak. Ugyanakkor több csapat is úgy gondolta, hogy klaszterezés szempontjából igen sok információt tartalmazhat a különböző dokumentumokban található tulajdonnevek [5].

[1] A korpusz szabadon elérhető a Creative Commons licenc alatt.

3 Jellemzőalapú személynév-egyértelműsítés

Alapvető hipotézisünk az, hogy az egyes személyeket leíró biográfiai jellemzők hasznosak lehetnek a klaszterező algoritmus számára. Például ha két honlapról is kiderül az illető születési helye és dátuma, és ezek megegyeznek, akkor majdnem biztosak lehetünk abban, hogy ugyanarról a személyről van szó. Ezért 16 különböző jellemzőosztály definiáltunk, és próbáltuk meg ezen osztályokba tartozó jellemzőket automatikusan kinyerni az egyes weboldalakból. Minden egyes dokumentumot az ezen jellemzőkből alkotott vektortérmodell reprezentált. Végül ezt a teret klasztereztük, és azonosítottunk az egyes személyekhez tartozó weboldalakat.

A jellemzők kinyerése során alapvetően a HTML-oldalak szöveges részeire fókuszáltunk, mivel úgy találtunk [9], hogy több oldal tartalmaz releváns információt szöveges részben, mint strukturált formában.

3.1 Előfeldolgozás

A rendszerünk bemenetéül egy személynévhez tartozó a Yahoo! kereső által visszaadott weboldalak szolgáltak. Mivel úgy találtuk, hogy a weboldalakon található hasznos információ nagyrészt azok szöveges részében fordul elő, ezért alapvetően az egyes oldalak szöveges bekezdéseire koncentráltunk. Ezáltal a különböző nyelvfeldolgozó eszközök számára zajos és nehezen feldolgozható elemeket elhagytuk.

A weboldalakon előforduló bekezdések azonosításához a magyarlancot [12] alkalmaztuk minden oldal DOM fájának elemeire. Amennyiben az oldalon található szövegrészlet hosszabb volt 60 karakternél és több mint egy igét tartalmazott, akkor azt bekezdésnek jelöltük. Néhány jellemzőt a saját tulajdonnév-felismerő [13] rendszerünkkel azonosítottunk, amelyet a HVG korpuszon tanítottunk.

3.2 B-Cubed kiértékelési metrika

A klaszterezés kiértékelési metrikájául az B-Cubed mérték [10] kiterjesztett változatát használtuk, követve a WePS 3 verseny hivatalos kiértékelési útmutatóját. Ebben az esetben pontosságot és fedést számolunk, ugyanakkor szükséges a helyesség kiterjesztése azokban az esetben, amikor egy dokumentumot több klaszterbe is besorolunk. Ezért definiáltuk a többszörös pontosságot és fedést:

$$\textit{Többszörös fedés}(e,e') = \frac{Min(|C(e) \cap C(e')|, |L(e) \cap L(e')|)}{|L(e) \cap L(e')|}$$

$$\textit{Többszörös pontosság}(e,e') = \frac{Min(|C(e) \cap C(e')|, |L(e) \cap L(e')|)}{|C(e) \cap C(e')|}$$

Ebben az esetben e és e' két különböző elem, míg L(e) az e elemhez tartozó kategóriákat, C(e) pedig az e-hez tartozó klasztereket jelöli. Többszörös pontosságot csak abban az esetben használtunk, amennyiben e és e' klasztereket osztott meg, továbbá

többszörös fedést, amennyiben e és e' kategóriákat osztott meg. Előző értéke akkor volt maximális (1), amennyiben a megosztott kategóriák száma kevesebb vagy egyenlő volt, mint a megosztott klaszterek száma. Ugyanakkor értéke akkor volt minimális (0), ha a két elem nem osztott meg egy kategóriát sem. A többszörös fedés értéke akkor volt maximális, amikor a megosztott klaszterek száma kevesebb vagy egyenlő volt a megosztott kategóriák számával, ha pedig két elem nem osztott meg egy klasztert sem, minimális értéket eredményezett.

3.3 Jellemzőkinyerés

Néhány a WePS 2 versenyen résztvevő jellemzőkinyerő rendszerhez hasonlóan a saját megközelítésünk is két fontos lépésre épült: lehetségesjellemző-kinyerési és jellemzőverifikációs modulra. Tehát először kinyertük a lehetséges jellemzőket a bekezdésekből, majd kiválasztottuk ezek közül a végsőket.

A jellemzőkinyerés részprobléma megoldása során szükségesnek tűnt a jellemzőosztályok kategorizálása. Az azonos logikai osztályba tartozókat csoportosítottuk. Így például azonos csoportba kerültek az *egyéb név*, *családtag* és a *mentor*, hiszen ezek mind egyes személyek nevei. Ugyanakkor egy hierarchikus rendszert definiáltunk az összetartozó jellemzőkön belül. Így egy nevet csak akkor jelöltünk *mentornak*, amennyiben az nem volt sem *családnév* és *egyéb név* sem. A jellemzők különböző csoportosításai az első táblázatban láthatók.

1. táblázat: A jellemzők csoportosítása.

Név	Elérhetőség	Szervezetek
családtag	e-mail	iskola
egyébnév	weboldal	díj
mentor	telefonszáma	affiliáció
	fax	

A továbbiakban kitérünk az egyes jellemzők azonosításának részleteire.

Születési dátum: amennyiben a szótövesítés után egy bekezdés tartalmazta a *születik*, *születési dátum* stb kifejezések bármelyikét, akkor lehetséges dátumokat kerestünk ezen szavak környezetében. Ehhez egy dátumvalidátort alkalmaztunk, amely 9 különböző reguláris kifejezés segítségével próbálja azonosítani a különböző formában megadott dátumokat.

Születési hely: amikor egy adott bekezdés szótövesített változata a *születik*, *szület*, *születni*, *szülőváros* kifejezések bármelyikét tartalmazta, akkor alkalmaztuk a saját, HVG korpusz földrajzi név osztályon tanított tulajdonnév-felismerő rendszerünket, hogy azonosítsuk a lehetséges szülőhelyeket. Végül egy frázist születési helynek jelöltünk, amennyiben azt a szülőhely-validátorunk elfogadta.

Szervezetek (*iskola*, *díj*, *affiliáció*): mivel úgy találtuk, hogy ezen jellemzők egyes szervezetek nevei, ezért ezeket egy csoportba soroltuk. Ugyancsak saját tulajdonnévfelismerő eszközünket alkalmaztuk, amely ebben az esetben a HVG korpusz szervezet osztályán lett tanítva. Amennyiben a tulajdonnév-felismerő rendszerünk talált egy

szervezetet a bekezdésekben, akkor azt először az iskola jellemző szempontjából vizsgáltuk. Amennyiben a kinyert szervezetnév megadott környékén előfordult valamely kulcskifejezés, mint például *diploma*, *oktatás*, *tudomány*, akkor azt lehetséges *iskolának* jelöltük. Egy ilyen kifejezést akkor fogadtunk véglegesen el, amennyiben az iskolavalidátorunk azt elfogadta. Ez abban az esetben történt meg, amennyiben az adott kifejezés minden szava nagybetűvel kezdődött, néhány kötőszót kivéve, mint például az *és,* továbbá tartalmaz néhány kulcskifejezést, úgymint *Iskola*, *Akadémia*, *Egyetem*, *Főiskola* stb. Amennyiben az adott kifejezést elvetettük, a továbbiakban *díj* jellemző szempontjából vizsgálódtunk. Így, amennyiben az aktuális kifejezés olyan kifejezések mellett fordul elő, mint *díj*, *nyer*, *év* stb. akkor azt potenciális *díj* attribútumként kezeltük. Egy ilyen kifejezést csak abban az esetben jelöltünk *díj* jellemzőnek, amennyiben azt egy általunk definiált díjvalidátor elfogadott. Ez akkor történt meg, ha az aktuális frázis minden szava nagybetűvel kezdődött, kivéve néhány kötőszót, úgymint az *és,* továbbá olyan kifejezéseket tartalmaz mint *díj*, *legjobb*, *játékos* stb. Amennyiben a potenciális szervezetnevet sem iskola, sem *díj* jellemzőként nem sikerült azonosítanunk, akkor azt végül *affiliáció*ként jelöltük.

Nevek (*családtag*, *egyéb név*, *mentor*): mivel ezen jellemzők mind valamilyen személyek nevei, ezért ezeket egy csoportba rendeltük. A név típusú attribútumok azonosítására szinten a saját fejlesztésű tulajdonnév-felismerő rendszerünket alkalmaztuk, ám ezúttal a HVG korpusz név osztálycímkéjén tanított modellt alkalmaztuk. A modell által kinyert személynévelemet *családtagnak* jelöltünk, amennyiben az valamilyen rokonságot kifejező szó környezetében fordult elő, mint például, *fia*, *apja* stb. (ezen kifejezések listáját a Wikipédia rokonság[2] szócikkéből gyűjtöttük). Azonban sok esetben ez a feltétel nem teljesült, így ekkor az aktuális potenciális nevet lehetséges *egyébnévként* kezeltük. Úgy gondoltuk, hogy egy adott személy nem ad meg egy másik nevet azonos számú szóval, (ez a megállapítás nem feltétlenül igaz becenevek esetén) ugyanakkor az *egyébnév* mindenképp tartalmazza az eredeti név legalább egy részét. Tehát ha az aktuális név Kovács István volt, Kovács Józsefet nem fogadtuk el egyébnévnek, míg Kovács T. Istvánt igen. Amennyiben egy nevet nem jelöltünk sem *családtag*, sem *egyébnévnek*, akkor azt végül a *mentor* jellemzőosztály szempontjából vizsgáltuk. Abban az esetben, ha az aktuális név néhány kulcskifejezés környékén fordult el, úgymint *edző*, *mentor* stb. akkor azt végül *mentor* osztályba soroltuk.

Titulus: manuálisan létrehoztunk egy 60 elemből álló listát, amely különböző tudományos fokozatokat, diplomákat tartalmaz. Amennyiben az aktuális név adott közelségében a lista egy elemét találtuk, akkor azt *titulus* jellemzőnek jelöltük.

Nemzetiség: összeállítottunk egy 371 elemből álló listát, mely különböző nemzeteket tartalmaz. Ekkor minden nemzetiséget megpróbáltunk kinyerni az oldalról, végül a leggyakoribbat jelöltük nemzetiség attribútumnak.

Amikor az elérhetőség jellemzőket próbáltuk azonosítani, akkor nem csak az oldalak bekezdéseit vizsgáltuk, hanem az egész oldalt, ugyanis úgy találtuk, hogy ezen típusú jellemzők bárhol előfordulhatnak a weboldalakon.

[2] http://hu.wikipedia.org/wiki/Rokonság

Telefonszám: amennyiben egy szövegrészlet tartalmazta a *tel*, *telefon*, stb. kifejezések egyikét, akkor a következő igen megengedő reguláris kifejezéssel kerestünk lehetséges telefonszámokat:

```
(((?[0-9+(][.()0-9s/-]{4,}[0-9])(? d{1,5})?)
```

Amennyiben volt találat, akkor egy általunk definiált validátor segítségével választottuk ki a telefonszámokat.

Fax: a telefonszámhoz hasonlóan jártunk el, mivel a két jellemző meglehetősen hasonló. Ugyanakkor ebben az esetben a *fax* szó környékén vizsgálódtunk.

E-mail: Úgy gondoltuk, hogy ha valaki közzéteszi az e-mail címét, az az esetek többségében egyben link is. Ezért elsősorban az olyan linkeket vizsgáltuk, amelyek a *mailto* tagot tartalmazták. Ezenkívül igen gyakori, hogy az e-mail cím tartalmazza a személy nevét, vagy annak egy darabját. Definiáltunk egy e-mailcím-validátort, amely abban az esetben fogadott egy e-mail címet, ha az tartalmazta a személy nevének karaktertrigramjainak valamelyikét. Ugyanakkor definiáltunk egy stoplistát is, amely olyan szavakat tartalmazott, mint például *webmaster*, *wiki*, *support* stb. Ezenkívül minden elfogadott e-mail címből kinyertük a *domain* címet, amit később az *internetcím* jellemzőnél használtuk fel.

Internetcím: Úgy találtuk, hogy a személyekhez köthető internetoldalak címe tartalmazzák az adott személynevet, vagy annak egy darabját. Ugyanakkor ezen attribútumok is jellemzően link formában fordulnak elő az egyes weboldalakon. Tehát az internetcím-validátorunk az olyan webcímeket fogadta el, amelyek tartalmazták az adott nevet, vagy annak egy részét, esetleg az e-mail címből kinyert domént.

3.4 Weboldalak klaszterezése

Az egy személyhez tartozó honlapok klaszterezése során úgy gondoltuk, hogy csupán a személyes információk alapján képesek vagyok a különböző emberekhez tartozó dokumentumokat osztályozni. Továbbá képesek vagyunk megállapítani, egy adott névhez tartozó dokumentumok hány különböző személyhez tartoznak. Ez az adat a klaszterezés szempontjából különösen fontos, hiszen a klaszterező algoritmusok többségéhez szükséges előre definiálni a klaszterek számát.

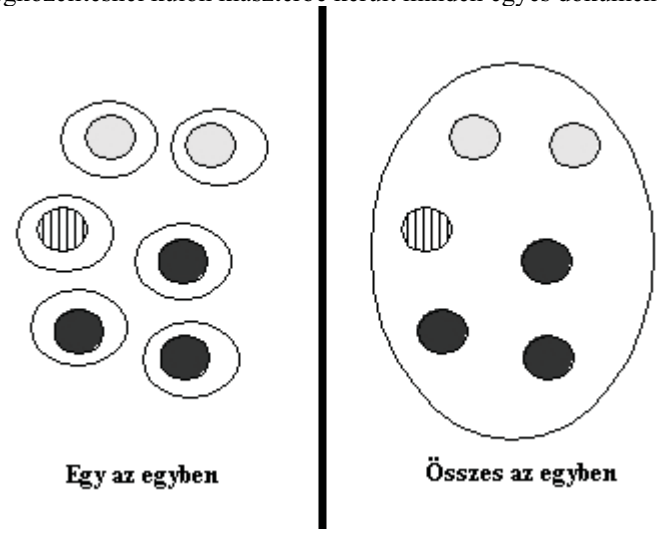
Az egyes jellemzőkre egy súlyozást definiáltunk. A leghasznosabb attribútumoknak az *internetcím*, *e-mail cím*, *telefonszám*, *fax* és az *egyéb név* bizonyult, ezért ezek 3 súlyt kaptak. Továbbá a *születési dátum* 2-es, míg a *születési hely*, *mentor*, *affiliáció*, *nemzetiség*, *családtag*, *iskola* és *díj* 1-es értéket kaptak. Ekkor minden dokumentumot a kinyert jellemzőkből álló vektor reprezentált. Ahhoz, hogy egy hatékony hasonlósági metrikát tudjunk meghatározni, előbb szükséges volt az egyes jellemzők normalizálása, egységes formára hozása. Ezért különböző szabályok és reguláris kifejezések segítségével egységesítettük azokat.

Az alapvetően weboldalakról kinyert személyes jellemzők segítségével végzett klaszterezés során egy alulról fölfelé történő heurisztikát alkalmaztunk. Ebben az esetben először minden dokumentum egy külön klaszterben van, majd a különböző klasztereket addig vonjuk össze iteratívan, amíg a megállási feltételt el nem érjük. Minden lépésben a leghasonlóbb klaszterek kerülnek összevonásra, ahol minden klaszter a centroidjával van reprezentálva, és két centroid közti távolság az őket leíró normalizált, súlyozott vektorok euklideszi távolsága. Az algoritmus számára a megál-

lási feltétel a legnagyobb hasonlósági mérték kevesebb mint 3-as mivolta. Tehát két klasztert abban az esetben nem vontunk össze, amennyiben a kettő közti hasonlósági mérték kisebb volt 3-nál.

Az alapvetően attribútumokat használó megoldás mellett, néhány alapmódszert is kipróbáltunk. Ekkor az egyes nevekhez tartozó dokumentumhalmazokat, a különböző dokumentumokból létrejövő vektortérmodell reprezentált. Ehhez a WEKA Java csomagban [11] található KMeans algoritmust is alkalmaztuk. Ugyanakkor ezen megközelítésnek, mint a klaszterező algoritmusok többségének, szükséges előre definiálni a klaszterek számát. Mivel az adott feladat során ez az érték nem ismert, ezért különböző heurisztikák segítségével próbáltuk meg megbecsülni azt. Az első esetben [Kmeans], az előzőekben már bemutatott, alulról fölfelé történő, jellemzőkön alapuló megközelítés által végeredményül kapott klaszterszámot adtuk meg. Másik esetben [Simple], a kiértékelő korpuszon, az egy névhez tartozó átlagos személyek számát (hét) adtuk meg. Végül a [Perfekt] esetben az annotátorok által meghatározott, adott névhez tartozó személyek számát kapta meg a KMeans algoritmus.

Ezen kívül még két egyszerű alapmegközelítést is adtunk, melyeket az 1. ábrán láthatunk. Az első esetben minden dokumentumot egy klaszterbe tettünk, míg az egy az egyben megközelítésnél külön klaszterbe került minden egyes dokumentum.



1. ábra. A két alapmegközelítés.

4 Kiértékelés

4.1 Korpusz

Rendszerünk kiértékelésére létrehoztunk egy magyar nevekhez tartozó weboldalkorpuszt, manuálisan annotált honlapokkal, amely elérhető a http://www.inf.u-

szeged.hu/rgai/nlp/homepagewsd weboldalon. Hogy eredményeink összevethetőek legyenek más nemzetközi eredményekkel, a tesztkorpuszt a meglévő korpuszokhoz hasonlóan hoztuk létre. A nevek közé több közéleti szereplő került, úgymint Csányi Sándor (OTP vezér és színész), továbbá Magyarországon igen gyakori nevek, mint például a Kovács István vagy a Szabó Zsófia. Ugyanakkor arra törekedtünk, hogy ezen gyakori nevek közt is szerepeljenek híres személyiségek, ahogy az első esetben a bokszoló, míg a másodikban a színésznő. Továbbá Schmitt Pál egy igazán érdekes kihívásnak ígérkezett, hiszen az élet különböző területein tölt be fontos pozíciókat, így a hozzá kapcsolódó weboldalak is igen eltérőek lehetnek.

A dokumentumhalmazban minden névhez a Yahoo![3] kereső által megadott első 100 találat került letöltésre, így a korpusz végül 960 weboldalt tartalmazott. Ezek közül összesen 572 oldalt kötöttek az annotátorok egy adott személyhez, vagyis egy névhez átlagosan 57 oldal kapcsolódott. Ugyanakkor a különböző nevek esetén igen nagy eltérések vannak, hiszen míg Zrínyi Miklós esetében a találatok nagy többsége valamilyen intézményhez köthető, addig például Schmitt Pál esetében az oldalak többsége a konkrét személyhez tartozik. A 10 névhez összesen 120 különböző személyt azonosítottunk, de míg a Kovács István esetében 30 különböző egyén fordult elő, addig a Schmitt Pálhoz tartozó weboldalak alapvetően a köztársasági elnökhöz voltak köthetők.

4.2 Eredmények

A különböző megközelítések eredményei a második táblázatban láthatóak. Az algoritmusokat B-Cubed pontosság, fedés és az ezekből számított F-mértékkel értékeltük ki. A táblázatból kitűnik, hogy az általunk megadott algoritmus érte el a legjobb eredményt az adott korpuszon. Míg a klaszterező eljárások közül az érte el a legjobb eredményt, amikor megadtuk a klaszterek pontos számát. A másik két eljárás más-más pontosság és fedés mellett ért el azonos F-mértéket.

2. táblázat: Eredmények.

Megközelítés	BCubed pontosság	BCubed fedés	F-mérték
Jellemzők	0,59	0,64	0,59
All_In_One	0,43	0,84	0,50
Perfekt	0,59	0,37	0,43
Simple	0,52	0,38	0,36
kMeans	0,69	0,28	0,36
One_In_One	0,93	0,24	0,35

5 Konklúzió

Ebben a cikkben bemutattunk az első magyar nyelvű személynév-egyértelműsítő megközelítésünket, amely hatékonyan volt képes kezelni a problémát. A kiértékelés-

[3] www.yahoo.com

hez létrehoztuk az első magyar nyelvű személynév-egyértelműsítő korpuszt. Rendszerünk a weboldalak folyó szöveges részéből dolgozik és alapvetően a személyek bibliográfiai attribútumai alapján egyértelműsíti a személyneveket. 16 különböző attribútumosztályt definiáltunk, amelyeket automatikus eszközökkel nyerünk ki. A klaszterezés 0,59 B-cubed F-mértéket ér el, ami az angol nyelvre publikált korpuszokkal és algoritmusokkal összevethető eredmény.

Habár eredményeink jónak tekinthetőek, a rendszernek számos továbbfejlesztési iránya van, amelyeket a jövőben meg kívánunk valósítani. Ilyenek például a táblázatos részekből kinyerhető információkkal való kiegészítés, mélyebb szintaktikai információk figyelembevétele a validátoroknál, illetve a bekezdések fő alanyainak azonosítása (a hibák egy része annak a hipotézisnek a következménye, hogy az egy oldalon talált minden bibliográfiai adat az oldal tulajdonosáé).

Köszönetnyilvánítás

A kutatást – részben – a TEXTREND projekt (Jedlik Ányos program) keretében az NKTH támogatta.

Bibliográfia

1. Ide, N., Veronis, J.: Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. Computational Linguistics Vol. 24 No. 1 (1998) 1–40
2. Guha, V., Garg, A.: Disambiguating People in Search. In: Proceedings of the 13th World Wide Web Conference (WWW 2004). ACM Press (2004)
3. A leggyakoribb magyar családnevek: http://www.chem.elte.hu/departments/elmkem/baranyai/nevek.htm
4. Sekine, S., Artiles, J.: WePS 2 Evaluation Campaign: Overview of the Web People Search Attribute Extraction Task. In: 2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference (2009)
5. Artiles, J., Gonzalo, J., Sekine, S.: The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. In: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). Association for Computational Linguistics, Prague, Czech Republic (2007) 64–69
6. Etzioni, O., Cafarella, M., Downey, D., Popescu, A. M., Shaked, T., Soderl, S., Weld, D. S., Yates, E. Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence Vol. 165 (2005) 91–134
7. Cheng, X., Adolphs, P., Xu, F., Uszkoreit, H., Li, H.: Gossip galore – a selflearning agent for exchanging pop trivia. In: Proceedings of the Demonstrations Session at EACL 2009. Association for Computational Linguistics, Athens, Greece (2009) 13–16
8. Li, H., Xu, F., Uszkoreit, H.: A seeddriven bottom-up machine learning framework for 8 extracting relations of various complexity. In: Proceedings of ACL 2007, 45th Annual Meeting of the Association for Computational Linguistics. Prague, Czech Republic (2007)
9. Nagy, I., Farkas, R., Jelasity, M.: Researcher affiliation extraction from homepages. In: Proceedings of the NLPIR4DL ACL Workshop (2009) 1–9

10. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using the vector space model. In: Proceedings of the 17th international conference on computational linguistics. ACL (1998)
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations, Vol. 11, No. 1(2009)
12. Zsibrita J., Nagy I., Farkas R.: Magyar nyelvi elemző modulok az UIMA keretrendszerhez. In: VI. Magyar Számítógépes Nyelvészeti Konferencia (2009) 394–395
13. Szarvas, Gy., Farkas, R., Kocsor, A.: A Multilingual Named Entity Recognition System Using Boosting and C4.5 Decision Tree Learning Algorithms. In: The Ninth International Conference on Discovery Science 2006. LNAI 4265 (2006) 267–278